

# **ARCHITETTURA DEI SISTEMI ELETTRONICI**

## **LEZIONE N° 8**

- Reti logiche
- Reti logiche combinatorie
- Reti logiche sequenziali
- Simboli
- Concetto di ciclo
- Concetto di minimizzazione (funzione costo)
- Realizzazioni diverse della stessa funzione
- Teorema di SHENNON
- Implicanti, Inclusivi, Implicanti Principali
- Mappe di Karnaugh
- Sintesi ottima
- Esempio di minimizzazione
- Considerazioni su soluzioni diverse
- Tecniche strutturate di minimizzazione

A.S.E.

8.1

## **Richiami**

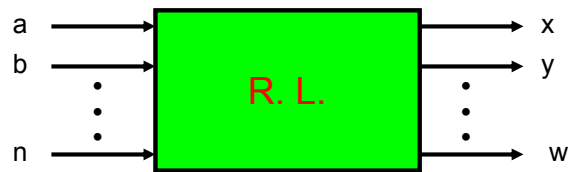
- Esempi di applicazione dei vari teoremi
- Passaggi da forma SP a PS e viceversa
- Insieme funzionalmente completo
- Funzione NAND
- Funzione NOR
- Funzioni AND, OR e NOT
- Funzioni NAND e NOR
- Funzioni XOR e XNOR

A.S.E.

8.2

## Reti Logiche

- Sistema elettronico che ha in ingresso segnali digitali e fornisce in uscita segnali digitali secondo leggi descrivibili con l'algebra Booleana



- R.L. è unidirezionale

A.S.E.

8.3

## Tipi di reti

- Reti **COMBINATORIE**
  - In qualunque istante le uscite sono funzione del valore che gli ingressi hanno in quell'istante
  - Il comportamento (uscite in funzione degli ingressi) è descritto da una tabella
- Reti **SEQUENZIALI**
  - In un determinato istante le uscite sono funzione del valore che gli ingressi hanno in quell'istante e i valori che hanno assunto precedentemente
  - La descrizione è più complessa
  - Stati Interni
  - Reti dotate di *MEMORIA*

A.S.E.

8.4

## Simboli

- Rete Logica => scomponibile in blocchi
- Blocchi base = simboli degli operatori elementari
- Rappresentazione delle funzioni logiche mediante schemi
- RAPPRESENTAZIONE SCHEMATICA

A.S.E.

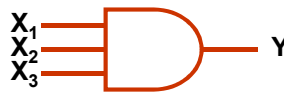
8.5

## Porte logiche

- Rappresentazione circuitale delle funzioni logiche

– AND

$$Y = X_1 \cdot X_2 \cdot X_3$$



– OR

$$Y = X_1 + X_2$$



– NOT

$$Y = \overline{X}$$



A.S.E.

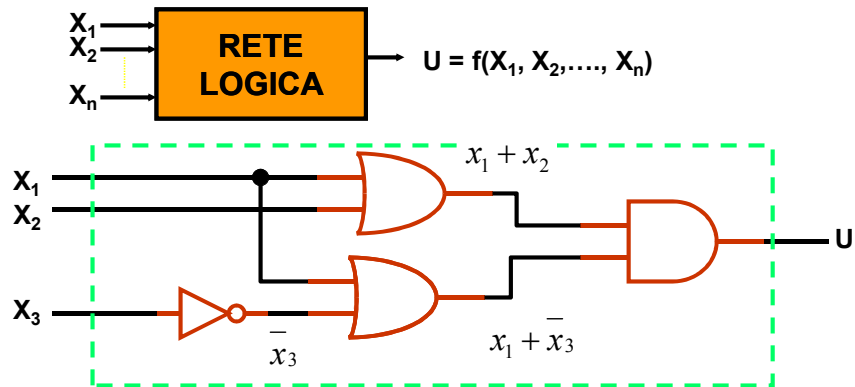
8.6

## Esempio

- Schema simbolico della funzione

$$U = f(X_1, X_2, \dots, X_n) = (X_1 + X_2) \cdot (X_1 + \overline{X_3})$$

- RETE LOGICA



A.S.E.

8.7

## Altre porte logiche

- NAND

X	Z	Y
0	0	1
0	1	1
1	0	1
1	1	0

$$Y = \overline{X \cdot Z}$$



- NOR

X	Z	Y
0	0	1
0	1	0
1	0	0
1	1	0

$$Y = \overline{X + Z}$$

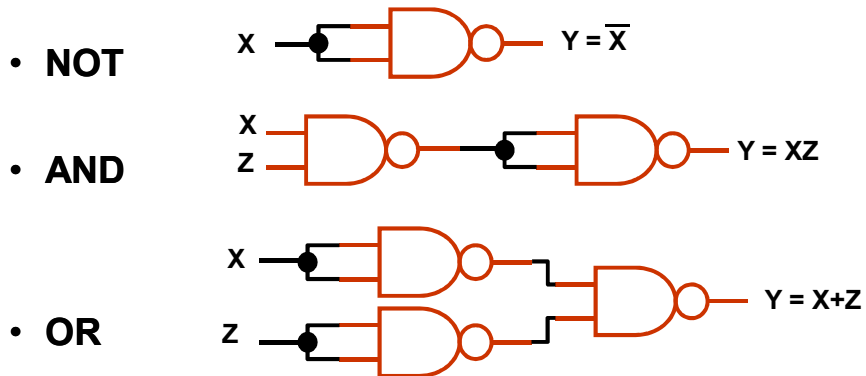


A.S.E.

8.8

## Proprietà della porta NAND (NOR)

- Utilizzando solamente porte NAND (NOR) è possibile realizzare qualunque rete logica



A.S.E.

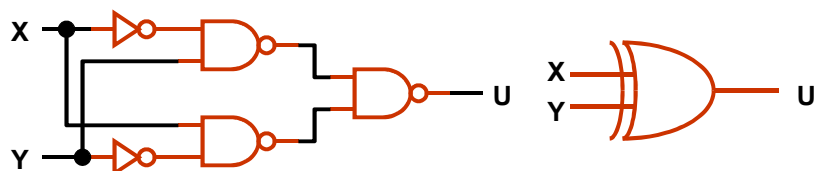
8.9

## OR Esclusivo

- Realizzazione dell'OR Esclusivo

X	Y	U
0	0	0
0	1	1
1	0	1
1	1	0

$$U = X \oplus Y = \bar{X}Y + X\bar{Y}$$



A.S.E.

8.10

## Ciclo

- **Definizione**
  - **Ciclo:** Percorso chiuso che attraversa  $k$  blocchi ( $k \geq 1$ ) tutti nella loro direzione di funzionamento
- **Osservazioni**
  - Tutte le reti viste sono prive di cicli
  - I blocchi base combinatori sono privi di cicli
  - Le funzioni descrivibili dalle tabelle di verità sono tutte prive di cicli (le uscite sono funzione dei solo ingressi)
- **Conclusione**
  - Tutte le reti logiche composte di blocchi combinatori e prive di cicli sono reti combinatorie

A.S.E.

8.11

## Sintesi di reti combinatorie

- **Sintesi**
  - data la descrizione ai terminali di una rete combinatoria
  - ottenere la struttura in blocchi logici e le relative interconnessioni
- **Osservazioni**
  - il funzionamento della rete deve essere possibile descriverlo mediante una tabella di verità
  - non esiste una sola realizzazione
  - per poter scegliere fra le varie soluzioni è necessario definire il parametro da ottimizzare
  - Funzione COSTO
  - (numero di blocchi base, ritardo ingresso uscita, uso di particolari blocchi, .....)
- **VEDERE ESEMPI SUCCESSIVI**

A.S.E.

8.12

## Esempio di funzione

- Data la funzione definita dalla Tabella di Verità:

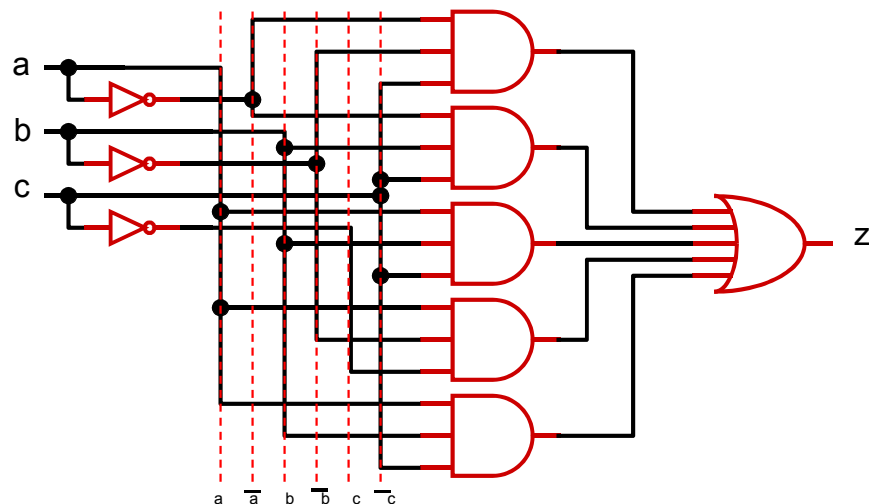
a	b	c	z	Si ha:
0	0	0	1	$z = \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$
0	0	1	0	$= \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$
0	1	0	1	$= \bar{c} + a \cdot \bar{b} = c \cdot (\overline{a \cdot b}) = c \cdot (\bar{a} + \bar{b}) = c \cdot \bar{a} + c \cdot \bar{b}$
0	1	1	0	$= \overline{(a + c)} + \overline{(c + b)}$
1	0	0	1	$z = (a + b + c) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + \bar{b} + \bar{c})$
1	0	1	1	
1	1	0	1	
1	1	1	0	

A.S.E.

8.13

## Schemi relativi 1

$$z = \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$$

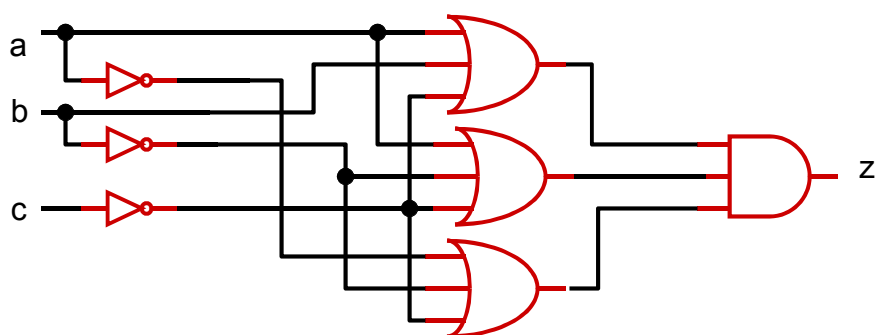


A.S.E.

8.14

## Schemi relativi 2

$$z = (a + b + \bar{c}) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + \bar{b} + \bar{c})$$

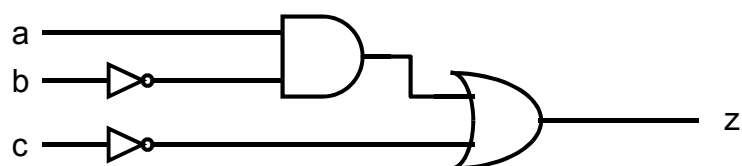


A.S.E.

8.15

## Schemi relativi 3

$$z = \bar{c} + a \cdot \bar{b}$$



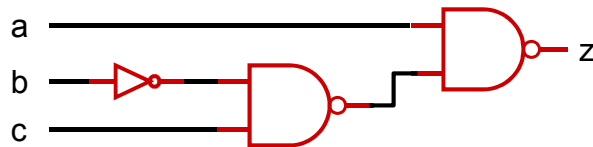
A.S.E.

8.16

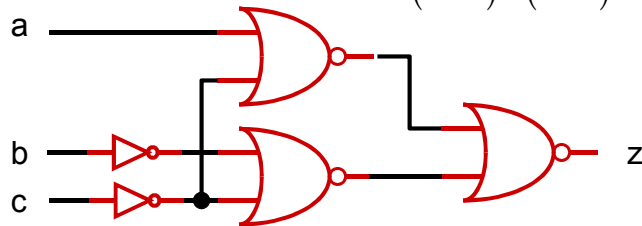


### Schemi relativi 4

$$z = c \cdot \overline{(a \cdot b)}$$



$$z = \overline{(a + c)} + \overline{(c + b)}$$



A.S.E.

8.17

### Teorema di espansione di Shannon

- Data la funzione

$$f(x_1, x_2, \dots, x_i, \dots, x_n)$$

- Vale la seguente uguaglianza

$$f(x_1, x_2, \dots, x_i, \dots, x_n) = x_i \cdot f(x_1, x_2, \dots, 1, \dots, x_n) + \overline{x_i} \cdot f(x_1, x_2, \dots, 0, \dots, x_n)$$

- Ovvero

$$f(x_1, x_2, \dots, x_i, \dots, x_n) = [x_i + f(x_1, x_2, \dots, 0, \dots, x_n)] \cdot [\overline{x_i} + f(x_1, x_2, \dots, 1, \dots, x_n)]$$

A.S.E.

10.18

## Esempio

- **Data la funzione**

$$f(w, x, y, z) = \overline{w}x + (wx + y)z$$

- **Risulta**

$$\begin{aligned} f(w, x, y, z) &= x \cdot f(w, 1, y, z) + \overline{x} \cdot f(w, 0, y, z) \\ &= x[\overline{w} \cdot 1 + (w \cdot 1 + y)z] + \overline{x}[\overline{w} \cdot 0 + (w \cdot 0 + y)z] \\ &= x[\overline{w} \cdot 0 + (w \cdot 1 + y)z] + \overline{x}[\overline{w} \cdot 1 + (w \cdot 0 + y)z] \\ &= x(w + y)z + \overline{x}(\overline{w} + yz) \end{aligned}$$

A.S.E.

10.19

## Osservazione

- **Applicando in modo iterativo il teorema di Shannon**

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &\equiv \overline{x_1} \cdot f(0, x_2, \dots, x_n) + x_1 \cdot f(1, x_2, \dots, x_n) = \\ &= \overline{x_1} \overline{x_2} \cdot f(0, 0, x_3, \dots, x_n) + \overline{x_1} x_2 \cdot f(0, 1, x_3, \dots, x_n) + \\ &\quad + x_1 \overline{x_2} \cdot f(1, 0, x_3, \dots, x_n) + x_1 x_2 \cdot f(1, 1, x_3, \dots, x_n) = \\ &= \overline{x_1} \overline{x_2} \overline{x_3} \cdot f(0, 0, 0, \dots, x_n) + \overline{x_1} \overline{x_2} x_3 \cdot f(0, 0, 1, \dots, x_n) + \\ &\quad + \overline{x_1} x_2 \overline{x_3} \cdot f(0, 1, 0, \dots, x_n) + \overline{x_1} x_2 x_3 \cdot f(0, 1, 1, \dots, x_n) + \\ &\quad + x_1 \overline{x_2} \overline{x_3} \cdot f(1, 0, 0, \dots, x_n) + x_1 \overline{x_2} x_3 \cdot f(1, 0, 1, \dots, x_n) + \\ &\quad + x_1 x_2 \overline{x_3} \cdot f(1, 1, 0, \dots, x_n) + x_1 x_2 x_3 \cdot f(1, 1, 1, \dots, x_n) \end{aligned}$$

- **Quindi il teorema di Shannon consente di ricavare sempre la forma SP**

A.S.E.

10.20

## Esempio

- Data la funzione

$$f(x, y, z) = \overline{x}\overline{y} + (x + y)z$$

- Risulta

$$\begin{aligned} f(w, x, y, z) &= \overline{x}\overline{y}z \cdot (\overline{00} + (0+0)0) + \overline{x}\overline{y}z \cdot (\overline{00} + (0+0)1) + \overline{x}\overline{y}z \cdot (\overline{01} + (0+1)0) + \\ &+ \overline{x}\overline{y}z \cdot (\overline{01} + (0+1)1) + \overline{x}\overline{y}z \cdot (\overline{10} + (1+0)0) + \overline{x}\overline{y}z \cdot (\overline{10} + (1+0)1) + \\ &+ \overline{x}\overline{y}z \cdot (\overline{11} + (1+1)0) + \overline{x}\overline{y}z \cdot (\overline{11} + (1+1)1) = \\ &= \overline{x}\overline{y}z \cdot (11 + (0+0)0) + \overline{x}\overline{y}z \cdot (11 + (0+0)1) + \overline{x}\overline{y}z \cdot (10 + (0+1)0) + \\ &+ \overline{x}\overline{y}z \cdot (10 + (0+1)1) + \overline{x}\overline{y}z \cdot (01 + (1+0)0) + \overline{x}\overline{y}z \cdot (01 + (1+0)1) + \\ &+ \overline{x}\overline{y}z \cdot (00 + (1+1)0) + \overline{x}\overline{y}z \cdot (00 + (1+1)1) = \\ &= \overline{x}\overline{y}z + \overline{x}\overline{y}z + \overline{x}\overline{y}z + \overline{x}\overline{y}z + \overline{x}\overline{y}z + \overline{x}\overline{y}z + \overline{x}\overline{y}z + \overline{x}\overline{y}z \end{aligned}$$

A.S.E.

10.21

## Implicanti

- Date due funzioni  $f_1$  e  $f_2$  di  $n$  variabili
- $f_1$  **implica**  $f_2$  se non c'è un assegnazione di valori alle  $n$  variabili tale che risulti  $f_1 = 1$  e  $f_2 = 0$
- Per funzioni booleane *completamente definite*
- Se  $f_1$  vale **1** anche  $f_2$  vale **1**
  - (Il fatto che  $f_1$  vale **1** **implica** che anche  $f_2$  vale **1**)
- Ovvero Se  $f_2$  vale **0** anche  $f_1$  vale **0**

A.S.E.

10.22

## Esempio 1

- Per  $f_1(x, y, z) = xy + yz$   
 $f_2(x, y, z) = xy + yz + \bar{x}z$

$x$	$y$	$z$	$f_1$	$f_2$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

A.S.E.

10.23

## Esempio 2

- Per  $f_3(x, y, z) = (x + y)(y + z)(\bar{x} + z)$   
 $f_4(x, y, z) = (x + y)(y + z)$

$x$	$y$	$z$	$f_3$	$f_4$
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	1
1	1	1	1	1

A.S.E.

10.24

## Osservazione

- Per una  $f$  funzione nella forma **SP**
  - Ogni termine di prodotto  $(x_1 \bar{x}_2 \cdots x_n)$  è implicante di  $f$
- Per una  $f$  funzione nella forma **PS**
  - La funzione  $f$  è implicante di ciascun termine di somma  $(x_1 + \bar{x}_2 + \cdots + x_n)$

A.S.E.

10.25

## Inclusione

- Dati due termini di prodotto  $p_1$  e  $p_2$ 
  - $p_1$  **include**  $p_2$  se e solo se tutti i letterali di  $p_2$  sono presenti in  $p_1$
- Dati due termini di somma  $s_1$  e  $s_2$ 
  - $s_1$  **include**  $s_2$  se e solo se tutti i letterali di  $s_2$  sono presenti in  $s_1$
- Se  $p_1$  include  $p_2$  allora  $p_1$  implica  $p_2$
- Se  $s_1$  include  $s_2$  allora  $s_2$  implica  $s_1$

A.S.E.

8.26

## Esempio

- Il termine di prodotto  $p_1 = \overline{x}\overline{y}\overline{z}$
- **Include** il termine di prodotto  $p_2 = x\overline{y}$
- Quindi  $p_1$  **implica**  $p_2$
  
- Il termine di somma  $s_1 = x + \overline{y} + \overline{z}$
- **Include** il termine di somma  $s_2 = x + \overline{z}$
- Quindi  $s_2$  **implica**  $s_1$

A.S.E.

10.27

## Implicanti principali

- Osservazioni
  - Tutti i termini di prodotto di una funzione booleana, nella forma SP, sono implicati della funzione
  - Tutti i mintermini di una funzione sono implicanti
- Un termine di prodotto che è implicante di una funzione è detto **Implicante Principale** se non include nessun altro implicante della funzione con un numero minore di letterali

A.S.E.

8.28

## Esempio

- Per la funzione definita dalla tabella di verità
- Sono implicanti di  $f$

$$\overline{\overline{xyz}}, \overline{x}, \overline{xyz}, \overline{yz}$$

- I termini  $\overline{\overline{xyz}}$  e  $\overline{xyz}$   
non sono implicanti principali

- I termini  $\overline{x}$  e  $\overline{yz}$   
sono implicanti principali  
(  $\overline{xyz}$  include  $\overline{x}$  ,  $\overline{xyz}$  include  $\overline{yz}$

$x$	$y$	$z$	$f$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

A.S.E.

10.29

## Sintesi ottima

- È necessario definire una funzione **COSTO** da minimizzare
- Definiti *letterali* le variabili dirette o complementate presenti in una funzione
- Date due forme diverse della stessa funzione
- La forma “A ” ha un costo minore della funzione “B ” se A contiene meno letterali.
- Minimizzare una funzione vuol dire trovare la forma con meno letterali
- Si possono definire altre funzioni COSTO in funzione della tecnologia realizzativa

A.S.E.

8.30

## Mappe di Karnaugh 1

- Tecnica tabulare di descrizione delle reti combinatorie
- Struttura a matrice
- Esempi

		b	
		0	1
a	0	$f(0,0)$	$f(0,1)$
	1	$f(1,0)$	$f(1,1)$

2 variabili

		b, c			
		00	01	11	10
a	0	$f(0,0,0)$	$f(0,0,1)$	$f(0,1,1)$	$f(0,1,0)$
	1	$f(1,0,0)$	$f(1,0,1)$	$f(1,1,1)$	$f(1,1,0)$

3 variabili

- si riportano solo gli “0” o solo gli “1”

A.S.E.

8.31

## Adiacenza

- Una combinazione delle variabili d'ingresso è detta **logicamente adiacente** a un'altra se le due combinazioni sono differenti solo in corrispondenza di un solo bit
- Nelle mappe, l'ordine delle combinazioni delle variabili è scelto in modo tale che due combinazioni **geometricamente adiacenti** siano anche **logicamente adiacente**

A.S.E.

8.32



## Mappe di Karnaugh 2

- **4 variabili**

		c d			
		00	01	11	10
a b	00	f(0000)	f(0001)	f(0011)	f(0010)
	01	f(0100)	f(0101)	f(0111)	f(0110)
	11	f(1100)	f(1101)	f(1111)	f(1110)
	10	f(1000)	f(1001)	f(1011)	f(1010)

- due colonne *adiacenti* differiscono per una sola variabile
- due righe *adiacenti* differiscono per una sola variabile
- la prima e l'ultima colonna sono adiacenti
  - La mappa è scritta su un cilindro verticale
- la prima e l'ultima riga sono adiacenti
  - La mappa è scritta su un cilindro orizzontale (ovvero la mappa sta su un toroide)

A.S.E.

8.33

## Mappe di Karnaugh 3

- **5 variabili**

		c d			
		00	01	11	10
a b	00	a	z		
	01	a x			
	11			y	
	10	b			b

e = 0

		c d			
		00	01	11	10
a b	00		c z	c	e
	01	d x			
	11	d		y	
	10				e

e = 1

- Le caselle con la stessa lettera sono adiacenti
- Attenzione alle caselle con lettere in **rosso** SONO ADIACENTI

A.S.E.

8.34

## Esempio

- Per la funzione prima trovata si ha

a	b	c	z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

<b>a \ b, c</b>		00	01	11	10
		0	1		
	1	1	1		1

<b>a \ b, c</b>		00	01	11	10
0			0	0	
1				0	

A.S.E.

8.35

## Osservazioni

- Data una funzione di “n” variabili
- Ogni casella della mappa corrisponde a un mintermine della funzione (prodotto di “n” termini)
- Due caselle adiacenti danno luogo a un prodotto di (n-1) termini
- Quattro caselle adiacenti danno luogo a un prodotto di (n-2) termini
- Otto caselle adiacenti danno luogo a un prodotto di (n-3) termini

A.S.E.

8.36

## Esempio 1

- Funzione “ $f$ ” di 4 variabili

$ab \backslash cd$	00	01	11	10
00	$\overline{a}\overline{b}\overline{c}\overline{d}$	$\overline{a}\overline{b}cd$	$\overline{a}b\overline{c}\overline{d}$	$\overline{a}bcd$
01	$\overline{a}b\overline{c}\overline{d}$	$\overline{a}bcd$	$a\overline{b}\overline{c}\overline{d}$	$a\overline{b}cd$
11	$ab\overline{c}\overline{d}$	$ab\overline{c}d$	$ab\overline{c}\overline{d}$	$ab\overline{c}d$
10	$ab\overline{c}\overline{d}$	$ab\overline{c}d$	$ab\overline{c}\overline{d}$	$ab\overline{c}d$

- La forma canonica SP si ottiene sommando le caselle dove  $f$  vale “1”

A.S.E.

8.37

## Esempio 2

- Data la funzione definita dalla seguente mappa:

$ab \backslash cd$	00	01	11	10
00	1	1		
01			1	
11	1	1		
10	1	1		

- si ha:

$$\begin{aligned}
 z &= \langle \overline{a}\overline{b}\overline{c}\overline{d} + \overline{a}\overline{b}cd \rangle + \overline{a}bcd + \langle a\overline{b}\overline{c}\overline{d} + a\overline{b}cd \rangle + \langle ab\overline{c}\overline{d} + ab\overline{c}d \rangle = \\
 &= \overline{a}\overline{b}\overline{c} \cdot (\overline{d} + d) + \overline{a}bcd + a\overline{b}\overline{c} \cdot (\overline{d} + d) + a\overline{b}\overline{c} \cdot (\overline{d} + d) = \\
 &= \overline{a}\overline{b}\overline{c} + \overline{a}bcd + \langle a\overline{b}\overline{c} + a\overline{b}\overline{c} \rangle = \overline{a}\overline{b}\overline{c} + \overline{a}bcd + a\overline{c} \cdot (b + \overline{b}) = \\
 &= \overline{a}\overline{b}\overline{c} + \overline{a}bcd + a\overline{c}
 \end{aligned}$$

A.S.E.

8.38

## Definizione

- Il prodotto “ $p$ ” si definisce implicante della funzione “ $f$ ” se  $p$  e  $f$  valgono “1” per la stessa configurazione degli ingressi
- I mintermini della funzione sono tutti implicanti della funzione
- Una funzione si può sempre scrivere come somma di implicanti
- Una casella delle mappe di Karnaugh è un implicante di ordine 1 (0) [1]
- Due caselle adiacenti sono un implicante di ordine 2 (1) [2]
- Quattro caselle adiacenti sono un implicante di ordine 3 (2) [4]
- Otto caselle adiacenti sono un implicante di ordine 4 (3) [8]
- L'espressione di un implicante si ricava direttamente dalle mappe di Karnaugh

A.S.E.

8.39

## Esempio

- Per la funzione prima vista si ha:

$$\begin{aligned}
 z &= \langle \overline{a}\overline{b}c\overline{d} + \overline{a}b\overline{c}\overline{d} \rangle + \overline{a}b\overline{c}d + \langle \overline{a}b\overline{c}\overline{d} + \overline{a}b\overline{c}d \rangle + \langle \overline{a}b\overline{c}\overline{d} + \overline{a}b\overline{c}d \rangle = \\
 &= \overline{a}\overline{b}\overline{c} \cdot (\overline{d} + d) + \overline{a}b\overline{c}d + \overline{a}b\overline{c} \cdot (\overline{d} + d) + \overline{a}b\overline{c} \cdot (\overline{d} + d) = \\
 &= \overline{a}\overline{b}\overline{c} + \overline{a}b\overline{c}d + \langle \overline{a}b\overline{c} + \overline{a}b\overline{c} \rangle = \overline{a}\overline{b}\overline{c} + \overline{a}b\overline{c}d + \overline{a}\overline{c} \cdot (b + \overline{b}) = \\
 &= \overline{a}\overline{b}\overline{c} + \overline{a}b\overline{c}d + \overline{a}\overline{c}
 \end{aligned}$$

Implicante di “z”

Implicante di ordine 2

Implicante di ordine 3

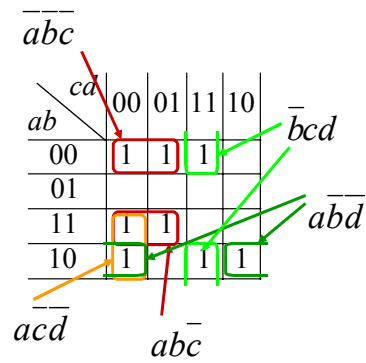
Implicante di ordine 1

A.S.E.

8.40

## Esempio

- Esempio di implicantanti di ordine 2

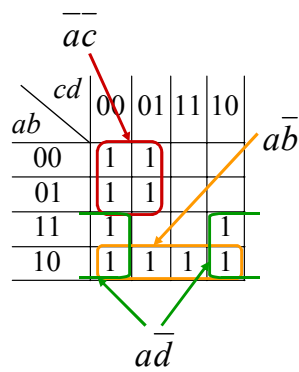


A.S.E.

8.41

## Esempio

- Esempio di implicantanti di ordine 3



A.S.E.

8.42

## Esempio

- Esempio di implicantanti di ordine 4

	$\bar{c}$				
	$c\bar{a}$	00	01	11	10
$ab$	00	1	1	1	1
	01	1	1		
	11	1	1		
	10	1	1	1	1

A.S.E.

8.43

## Definizione

- **Richiamo**
  - Una funzione si può sempre scrivere come somma di implicantanti
$$f = p_1 + p_2 + \dots + p_n$$
- Un implicants  $p^*$  si dice **implicants principale** se non esiste nessun altro implicants  $p'$  tale che  $p'$  **copra**  $p^*$
- Per ogni funzione  $f$  esiste almeno un insieme di implicants principali tale che  $f$  può essere espressa come somma di soli implicants principali

A.S.E.

8.44

## Esempio

- Per la funzione prima vista :

cd \ ab	00	01	11	10
00	1	1		
01			1	
11	1	1		
10	1	1		

- si ha:

$$z = \overline{a}\overline{b}c + \overline{a}bcd + ac + abc$$

- L'implicane **verde** non è principale

A.S.E.

8.45

## Ottimizzazione mediante le Mappe di Karnaugh

- **Passo 1**
  - individuare sulla mappa *tutti* gli implicant di ordine superiore possibile che coprono tutta la funzione
- **Passo 2**
  - Scegliere un insieme *più piccolo possibile* di implicant principali che coprono la funzione
- **NOTA**
  - L'ottimizzazione si fa per ispezione visiva

A.S.E.

8.46

## Esempio

- Per la funzione prima vista :

$ab \backslash cd$	00	01	11	10
00	1	1		
01			1	
11	1	1		
10	1	1		

$ab \backslash cd$	00	01	11	10
00	1	1		
01			1	
11	1	1		
10	1	1		

$ab \backslash cd$	00	01	11	10
00	1	1		
01			1	
11	1	1		
10	1	1		

- si ha:

$$z = \overline{b}c + a\overline{c} + abcd$$

- La scelta 3 da luogo ad una funzione migliore delle altre

A.S.E.

8.47

## Esempio di minimizzazione

- Data la funzione precedentemente vista:

a	b	c	z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Si ha:

$a \backslash b, c$	00	01	11	10
0	1			1
1	1	1		1

$$z = \overline{c} + a \cdot \overline{b}$$

A.S.E.

8.48



## Condizioni non specificate

a	b	c	d	z
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	–
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	–
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	–
1	1	1	1	0

» Può capitare che in particolari applicazioni alcune configurazioni degli ingressi non si possano verificare, quindi l'uscita per tali uscite non è specificata (*Don't-Care Conditions*)

cd \ ab	00	01	11	10
00	1	1	–	1
01			1	
11	1	1		–
10	–	1		

$z = \overline{a}\overline{b}\overline{c} + \overline{a}\overline{b}\overline{d} + a\overline{b}\overline{c} + a\overline{b}\overline{d} + \overline{a}bcd$   
 $z = \overline{a}b + \overline{a}cd + a\overline{c}$

» Se i *don't care* si considerano “0” si ottiene la prima funzione  
 » Se alcuni *don't care* si considerano “1” si ottiene la seconda funzione

A.S.E. 8.49

## Un cattivo esempio

a	b	c	d	z	w	u
0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	0	1	0	0	1	1
0	0	1	1	0	0	0
0	1	0	0	1	0	1
0	1	0	1	1	1	0
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	1	1	0
1	0	1	1	1	0	1
1	1	0	0	0	0	0
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	0	0	0

cd \ ab	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

$u = \overline{z}\overline{w} + \overline{z}w = z \oplus w$   
 $z = a\overline{b} + \overline{a}b = a \oplus b$   
 $w = c\overline{d} + \overline{c}d = c \oplus d$   
 $u = \overline{a}\overline{b}cd + \overline{a}b\overline{c}d + \overline{a}b\overline{c}\overline{d} + \overline{a}bcd + a\overline{b}\overline{c}d + a\overline{b}c\overline{d} + a\overline{b}cd + \overline{a}bcd$   
 $u = a \oplus b \oplus c \oplus d$

A.S.E. 8.50

## **Tecniche strutturate**

- Il procedimento di sintesi per “ispezione visiva” si può utilizzare fino a 4 ÷ 5 variabili
- Il procedimento di sintesi per “ispezione visiva” può essere anche descritto come processo formale strutturato
- Metodo di Quine McCluskey
- Può essere tradotto in un programma
- La complessità del programma cresce in modo esponenziale con l'aumentare delle variabili
- I programmi attuali usano tecniche euristiche

A.S.E.

8.51

## **Conclusioni**

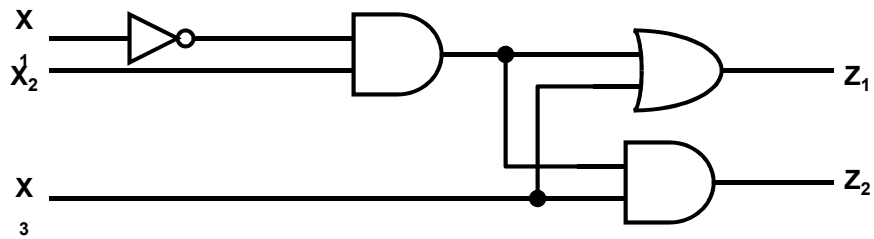
- Funzione XOR
- Enumerazione di funzioni
- Reti logiche
- Reti logiche combinatorie
- Reti logiche sequenziali
- Simboli
- Esempi
- Concetto di ciclo
- Realizzazioni diverse della stessa funzione
- Mappe di Karnaugh
- Implicanti
- Implicanti principali

A.S.E.

8.52

## Quesiti

- Ricavare le funzioni logiche di  $Z_1$  e  $Z_2$

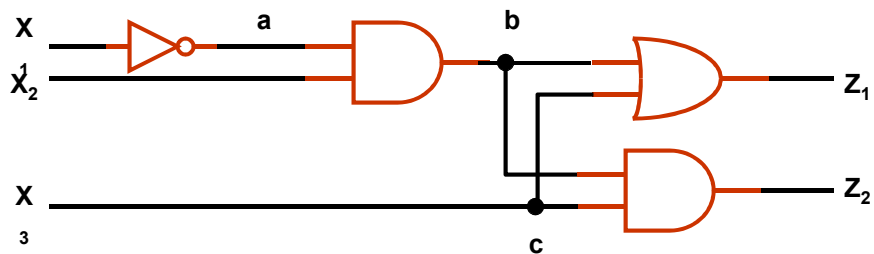


A.S.E.

8.53

## Suggerimenti

- Scrivere la tabella di verità comprensiva delle funzioni intermedie “a”, “b” e “c”



A.S.E.

8.54